

Interpreting Explain Plan Output

John Mullins

jmullins@themisinc.com

www.themisinc.com

www.themisinc.com/webinars

Presenter

- John Mullins
- Themis Inc. (jmullins@themisinc.com)
- 30+ years of Oracle experience
- Oracle Certified Professional DBA
- Certified Technical Trainer
- Over 250 classes taught

Themis Inc.

- More than 18 years in the industry
- Committed to your training
- Courses:
 - DB2, SQL Server, Oracle, Unix, Linux, Java, Web Development, .NET and many more
- www.themisinc.com

Related / Upcoming Themis Classes

- Oracle SQL Optimization
- Introduction to Oracle SQL
- Oracle Advanced SQL

Webinar Objectives

- Gain an understanding of how to interpret the output generated by the Explain Plan utility

Explain Plan Overview

- An explain plan is a representation of the access path that is taken when a query is executed within Oracle.
- The explain plan is produced by the optimizer. Once the access path has been decided upon it is stored in the library cache memory structure together with the statement itself.

Reading Explain Plan Output

- Read from the inside out
- Read from the top down

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		10	570	7 (15)	00:00:01
* 1	HASH JOIN		10	570	7 (15)	00:00:01
2	NESTED LOOPS					
3	NESTED LOOPS		10	380	4 (0)	00:00:01
* 4	TABLE ACCESS FULL	DEPARTMENT	1	16	3 (0)	00:00:01
* 5	INDEX RANGE SCAN	EMP_DEPARTMENT_IX	10		0 (0)	00:00:01
6	TABLE ACCESS BY INDEX ROWID	EMPLOYEE	10	220	1 (0)	00:00:01
7	VIEW	index\$_join\$_114	107	2033	3 (34)	00:00:01
* 8	HASH JOIN					
9	INDEX FAST FULL SCAN	EMP_NAME_IX	107	2033	1 (0)	00:00:01
10	INDEX FAST FULL SCAN	EMP_EMP_ID_PK	107	2033	1 (0)	00:00:01

The order of the operations is 4, 5, 3, 6, 2, 9, 10, 8, 7, 1, 0.

What to Look For...

- FULL Table Scans
- Join Methods, Join Order
- Index Access Methods
- Filters
- Parallel Operations
- Partition Processing
- Dynamic Statistics
- Cost

...What to Look For...

■ FULL Table Scans

- The entire table is read up to the high water mark (HWM). The HWM marks the last block in the table that has ever had data written to it. If you have deleted all the rows in a table you will still read up to the HWM.

...What to Look For...

- Join Methods, Join Order
 - Nested Loop
 - Hash Join
 - Sort Merge Join
 - Cartesian Join

...What to Look For...

■ Nested Loop

- First we return all the rows from row source 1 (Driving Table), typically the smaller of the two row sources.
- Then we probe row source 2 (Inner Table) once for each row returned from row source 1.
- Good for joining smaller row sources.
- Best used with indexed foreign key columns.

...What to Look For...

■ Hash Join

- Smallest row source is chosen and used to build a hash table (in memory) and a bitmap.
- The second row source is hashed and checked against the hash table looking for joins. The bitmap is used as a quick lookup to check if rows are in the hash table.
- Good for joining larger row sources.
- Needs PGA memory.

...What to Look For...

■ Sort Merge Join

- Rows are produced by row source 1 and are then sorted.
- Rows from row source 2 are then produced and sorted by the same sort key as Row Source 1.
- Row source 1 and 2 are NOT accessed concurrently. Sorted rows from both sides are then merged together.
- Needs PGA Memory.

...What to Look For...

■ Cartesian Join

- Every row from one row source is joined to every row from the second row source.
- Usually the result of a poorly written join.

...What to Look For...

- Index Access Methods
 - Index Range Scan
 - Index Unique Scan
 - Index Full Scan
 - Index Fast Full Scan
 - Index Skip Scan

...What to Look For...

■ Index Range Scan

- Method for accessing multiple column values.
- A non-unique index may return multiple values for the predicate `col1 = n` and will use an index range scan.

TABLE ACCESS BY ROWID EMP
INDEX RANGE SCAN EMP_IX1

...What to Look For...

- Index Unique Scan
 - Method for looking up a single key value via a unique index. Always returns a single value.

TABLE ACCESS BY ROWID EMP
INDEX UNIQUE SCAN EMP_IX0

...What to Look For...

■ Index Full Scan

- In certain circumstances it is possible for the whole index to be scanned where no constraining predicates are provided for a table.
- An index full scan will perform single block i/o's and so it may prove to be inefficient.

SELECT STATEMENT

INDEX FULL SCAN EMP_IX2

...What to Look For...

■ Index Fast Full Scan

- Scans all the block in the index Rows are not returned in sorted order.
- Uses multiblock i/o and can be executed in parallel.

```
SELECT STATEMENT  
  INDEX FAST FULL SCAN EMP_IX3
```

...What to Look For...

■ Index Skip Scan

- The optimizer can perform skip scans to retrieve rows for values that do not use the prefix of a concatenated index.
- Initiated by probing the index for distinct values of the prefix column. Each of these distinct values is then used as a starting point for a regular index search.

```
SELECT STATEMENT  
  INDEX SKIP SCAN EMP_IX4
```

...What to Look For...

■ Filters

- Restrictions in the WHERE clause.
- Optimizer will try to filter rows before performing joins and this is the goal.

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		5	110	6 (0)	00:00:01
* 1	HASH JOIN		5	110	6 (0)	00:00:01
* 2	TABLE ACCESS FULL	DEPT	1	13	3 (0)	00:00:01
* 3	TABLE ACCESS FULL	EMP	5	45	3 (0)	00:00:01

Predicate Information (identified by operation id):

- 1 - access("D"."DEPTNO"="E"."DEPTNO")
- 2 - filter("D"."DEPTNO"=20)
- 3 - filter("E"."DEPTNO"=20)

...What to Look For...

■ Parallel Operations

- Some SQL processing can be broken into separate parallel processing steps.
- Parallel processing can be hinted.
- Some operations that might be seen in the execution plan:
 - PX COORDINATOR
 - PX BLOCK ITERATOR
 - PX SEND
 - PX RECEIVE

...What to Look For...

■ Partition Processing

- Common practice to partition tables that are expected to contain a large volume of rows.
- The optimizer is partition-smart, and its plans should reflect this. (Partition Pruning)
- Partition access in the execution plan:
 - PARTITION LIST SINGLE
 - PARTITION LIST ALL

...What to Look For...

■ Dynamic Statistics

- Indicates object statistics are missing.

Oracle 12c Example:

```
-----  
| Id | Operation          | Name | Rows | Bytes | Cost (%CPU)| Time      |  
-----  
|  0 | SELECT STATEMENT   |      |    5 |   445 |      3   (0)| 00:00:01 |  
|*  1 |  TABLE ACCESS FULL| EMP  |    5 |   445 |      3   (0)| 00:00:01 |  
-----
```

Predicate Information (identified by operation id):

PLAN_TABLE_OUTPUT

1 - filter("DEPTNO">=20)

Note

- dynamic statistics used: dynamic sampling (level=2)

...What to Look For

■ Cost

- Represents the estimated resource usage for a plan. The optimizer's cost model accounts for the IO, CPU, and network resources that will be used by the query.
- Cost for an entire plan is indicated on line 0 of the explain plan output.
- Used to compare different plans for the same query.

Summary

- There are many things to look for in an execution plan for an SQL statement
- Questions?
 - jmullins@themisinc.com

Themis Oracle Classes

- Classes are offered in a public classroom, onsite at your location, or live on the internet
 - Oracle SQL Optimization
 - Oracle SQL, Oracle Advanced SQL
 - Oracle PL/SQL, Oracle Advanced PL/SQL
 - Oracle Database Performance Tuning for Administrators
 - And many more...

For More Information

- Visit the Themis web site www.themisinc.com
- John Caccavale
 - jcac@themisinc.com
 - 908-233-8900
- To get a copy of the presentation:
 - <http://www.themisinc.com/webinars>
- Thank you for attending. Have a good day.